A COMPUTATIONALLY EFFICIENT FEASIBLE SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM

Craig T. Lawrence and André L. Tits
Department of Electrical and Computer Engineering
and Institute for Systems Research
University of Maryland, College Park
College Park, Maryland 20742

October 7, 2000

Abstract

A Sequential Quadratic Programming (SQP) algorithm generating feasible iterates is described and analyzed. What distinguishes this algorithm from previous feasible SQP algorithms proposed by various authors is a reduction in the amount of computation required to generate a new iterate while the proposed scheme still enjoys the same global and fast local convergence properties. A preliminary implementation has been tested and some promising numerical results are reported.

1 Introduction

Consider the inequality-constrained nonlinear programming problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_j(x) \leq 0, \quad j = 1, \ldots, m, \end{array} \tag{P}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, m$, are continuously differentiable. Sequential Quadratic Programming (SQP) algorithms are widely acknowledged to be among the most successful algorithms available for solving (P). For an excellent recent survey of SQP algorithms, and the theory behind them, see [2].

Denote the feasible set for (P) by

$$X \stackrel{\Delta}{=} \{ \ x \in \mathbb{R}^n \ | \ g_j(x) \leq 0, \ j = 1, \ldots, m \ \}.$$

In [17, 8, 14, 15, 1], variations on the standard SQP iteration for solving (P) are proposed which generate iterates lying within X. Such methods are sometimes referred to as "Feasible SQP" (or FSQP) algorithms. It was observed that requiring feasible iterates has both algorithmic and application-oriented advantages. Algorithmically, feasible iterates are desirable because

- The QP subproblems are always consistent, i.e. a feasible solution always exists, and
- The objective function may be used directly as a merit function in the line search.

In an engineering context, feasible iterates are important because

- Often $f(x)$ is undefined outside of the feasible region X,
- Trade-offs between design alternatives (all requiring that "hard" constraints be satisfied) may be meaningfully explored, and
- The optimization process may be stopped after a few iterations, yielding a feasible point.

The last feature is critical for real-time applications, where a feasible point may be required before the algorithm has had time to "converge" to a solution.

An important function associated with the problem (P) is the *Lagrangian* $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, which is defined by

$$L(x, \lambda) \triangleq f(x) + \sum_{i=1}^{m} \lambda_i g_i(x).$$

Given a feasible estimate x of the solution of (P) and a symmetric matrix H that approximates the Hessian of the Lagrangian $L(x, \lambda)$, where λ is a vector of non-negative Lagrange multiplier estimates, the standard SQP search direction, denoted $d^0(x, H)$ or d^0 for short, solves of the Quadratic Program (QP)

$$\begin{array}{ll} \min & \frac{1}{2}\langle d^0, Hd^0\rangle + \langle \nabla f(x), d^0\rangle \\ \text{s.t.} & g_j(x) + \langle \nabla g_j(x), d^0\rangle \leq 0, \quad j = 1, \ldots, m. \end{array} \qquad QP^0(x, H)$$

Positive definiteness of H is often assumes as it ensures existence and uniqueness of such solution. With an appropriate merit function, line search procedure, Hessian approximation rule, and (if necessary) Maratos effect [13] avoidance scheme, the SQP iteration is known to be globally and locally superlinearly convergent (see, e.g., [2]).

A *feasible direction* at a point $x \in X$ is defined as any vector d in $\mathbb{R}^n$ such that $x + td$ belongs to X for all t in $[0, \bar{t}\,]$, for some positive $\bar{t}$. Note that the SQP direction d^0, a direction of descent for f, may not be a feasible direction at x, though it is at worst tangent to the active constraint surface. Thus, in order to generate feasible iterates in the SQP framework, it is necessary to "tilt" d^0 into the feasible set. A number of approaches have been considered in the literature for generating feasible directions and, specifically, tilting the SQP direction.

Early feasible direction algorithms (see, e.g., [27, 17]) were first-order methods, i.e. only first derivatives were used and no attempt was made to accumulate and use second-order information. Furthermore, search directions were often computed via linear programs instead of QPs. As a consequence, such algorithms converged linearly at best. Polak proposed several extensions to these algorithms (see [17], Section 4.4) which took second-order information into account when computing the search direction. A few of the search directions proposed by Polak could be viewed as tilted SQP directions (with proper choice of the matrices encapsulating the second-order information in the defining equations). Even with second-order information, though, it is not possible to guarantee superlinear convergence of these algorithms because no mechanism was included for controlling the *amount* of tilting.

A straightforward way to tilt the SQP direction is, of course, to perturb the right-hand side of the constraints in $QP^0(x, H)$. Building on this observation, Herskovits and Carvalho [8] and Panier and Tits [14] independently developed similar feasible SQP algorithms in which the size of the perturbation was a function of the norm of $d^0(x, H)$ at the current feasible point x. Thus, their algorithms required the solution of $QP^0(x, H)$ in order to *define* the perturbed QP. Both algorithms were shown to be superlinearly convergent. On the other hand, as a by-product of the tilting scheme, global convergence proved to be more elusive. In fact, the algorithm in [8] is not globally convergent, while the algorithm in [14] has to resort to a first-order search direction far from a solution in order to guarantee global convergence. Such a hybrid scheme could give slow convergence if a poor initial point is chosen.

The algorithm developed by Panier and Tits in [15], and analyzed under weaker assumptions by Qi and Wei in [20], has enjoyed a great deal of success in practice as implemented in the FFSQP/CFSQP [26, 12] software packages. We will refer to their algorithm throughout this paper as **FSQP**.

In [15], instead of directly perturbing $QP^0(x,H)$, tilting is accomplished by replacing d^0 with the convex combination $(1-\rho)d^0+\rho d^1$, where d^1 is an (essentially) arbitrary feasible descent direction. To preserve the local convergence properties of the SQP iteration, ρ is selected as a function $\rho(d^0)$ of d^0 in such a way that d approaches d^0 fast enough (in particular, $\rho(d^0)=O(\|d^0\|^2)$) as the solution is approached. Finally, in order to avoid the Maratos effect and guarantee a superlinear rate of convergence, a second order correction d^{C} (denoted $\tilde{d}$ in [15]) is used to "bend" the search direction. That is, an Armijo-type search is performed along the arc $x+td+t^2d^{\mathrm{C}}$, where d is the tilted direction. In [15], the directions d^1 and d^{C} are both computed via QPs but is is pointed out that d^{C} could instead be taken as the solution of a linear least squares problem without affecting the asymptotic convergence properties.

From the point of view of computational cost, the main drawback of algorithm **FSQP** is the need to solve three QPs (or two QPs and a linear least squares problem) at each iteration. Clearly, for many problems it would be desirable to reduce the number of QPs at each iteration while preserving the generation of feasible iterates as well as the global and local convergence properties. This is especially critical in the context of those large-scale nonlinear programs for which the time spent solving the QPs dominates that used to evaluate the functions.

With that goal in mind, consider the following perturbation of $QP^0(x,H)$. Given a point x in X, a symmetric positive definite matrix H, and a non-negative scalar η, let $(d(x,H,\eta),\gamma(x,H,\eta))$ solve the QP

$$\begin{array}{ll} \min & \frac{1}{2}\langle d, Hd\rangle+\gamma \\ \text{s.t.} & \langle\nabla f(x), d\rangle \leq \gamma, \\ & g_j(x)+\langle\nabla g_j(x), d\rangle \leq \gamma\cdot\eta, \quad j=1,\ldots,m \end{array} \qquad QP(x,H,\eta)$$

where γ is an additional, scalar variable.

The idea is that, away from KKT points of (P), $\gamma(x,H,\eta)$ will be negative and thus $d(x,H,\eta)$ will be a descent direction for f (due to the first constraint) as well as, if η is strictly positive, a feasible direction (due to the m other constraints). Note that when η is set to one the search direction is a special case of those computed in Polak's second-order feasible direction algorithms (again, see Section 4.4 in the book [17]). Further, it is not difficult to show that when η is set to zero, we recover the SQP direction, i.e. $d(x,H,0)=d^0(x,H)$. Large values of the parameter η, which we will call the *tilting parameter*, emphasize feasibility, while small values of η emphasize descent.

In [1], Birge, Qi, and Wei propose a feasible SQP algorithm based on $QP(x, H, \eta)$. Their motivation for introducing the right-hand-side constraint perturbation and the tilting parameters (they use a vector of parameters, one for each constraint) is, like ours, to obtain a feasible search direction. Specifically, motivated by the high cost of function evaluations in the application problems they are targeting, their goal is to ensure that a full step of one is accepted in the line search as early on as is possible (so that costly line searches are avoided for most iterations). To this end, their tilting parameters start out positive and, if anything, increase when a step of one is not accepted. A side-effect of such an updating scheme is that the algorithm cannot achieve a superlinear rate of convergence, as the authors point out in Remark 5.1 of [1].

In the present paper, our goal is to compute a feasible descent direction which approaches the true SQP direction fast enough so as to ensure superlinear convergence. Furthermore, we would like to do this with as little computation per iteration as possible. While computationally rather expensive, algorithm **FSQP** of [15] has the convergence properties and practical performance we seek. We thus start with reviewing its key features. For x in X, define

$$I(x) \stackrel{\Delta}{=} \{\ j \mid g_j(x) = 0\ \},$$

the index set of active constraints at x. In **FSQP**, in order for the line-search (with the objective function f used directly as the merit function) to be well-defined, and in order to preserve global and fast local convergence, the sequence of search directions $\{d_k\}$ generated by algorithm **FSQP** is constructed so that the following properties hold:

P1 $d_k = 0$ if x_k is a KKT point for (P),

P2 $\langle \nabla f(x_k), d_k \rangle < 0$ if x_k is not a KKT point,

P3 $\langle \nabla g_j(x_k), d_k \rangle < 0$, for all $j \in I(x_k)$ if x_k is not a KKT point, and

P4 $d_k = d_k^0 + O(\|d_k^0\|^2)$.

We will show in Section 3 that given any symmetric positive definite matrix H_k and non-negative scalar η_k, $d(x_k, H_k, \eta_k)$ automatically satisfies P1 and P2. Furthermore, it satisfies P3 if η_k is strictly positive. Ensuring that P4 holds requires a bit more care.

In the algorithm proposed in this paper, at iteration k, the search direction is computed via solving $QP(x_k, H_k, \eta_k)$ and the tilting parameter η_k is iteratively adjusted to ensure that the four properties are satisfied. The resultant algorithm will be shown to be locally superlinearly convergent and globally convergent without resorting to a first-order direction far from the solution. Further, the generation of a new iterate only requires the solution of one QP and two closely related linear least squares problems. In contrast with the algorithm presented in [1], our tilting parameter starts out positive and asymptotically approaches zero.

Recently there has been a great deal of interest in interior point algorithms for nonconvex nonlinear programming (see, e.g., [5, 6, 24, 4, 16, 23]). Such algorithms generate feasible iterates and typically only require the solution of linear systems of equations in order to generate new iterates. SQP-type algorithms, however, are often at an advantage over such methods in the context of applications where the number of variables is not too large but evaluations of objectives/constraint functions and of their gradients are highly time-consuming. Indeed, because these algorithms use quadratic programs as successive models, away from a solution, progress between (expensive) function evaluations is often significantly better than that achieved by algorithms making use of mere linear systems of equations as models.

In Section 2, we present the details of our new FSQP algorithm. In Section 3, we show that under mild assumptions our iteration is globally convergent, as well as locally superlinearly convergent. The algorithm has been implemented and tested and we show in Section 4 that the numerical results are quite promising. Some related issues are discussed in Section 5. Finally, in Section 6, we offer some concluding remarks and discuss some extensions to the algorithm which are currently being explored.

2 Algorithm

We begin by making a few assumptions that will be in force throughout.

Assumption 1: The set X is non-empty.

Assumption 2: The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \dots, m$, are continuously differentiable.

Assumption 3: For all $x \in X$ with $I(x) \neq \emptyset$, the set $\{\nabla g_j(x) \mid j \in I(x)\}$ is linearly independent.

A point $x^* \in \mathbb{R}^n$ is said to be a *Karush-Kuhn-Tucker (KKT)* point for the problem (P) if there exist scalars (*KKT multipliers*) $\lambda^{*,j}$, $j = 1, \dots, m$, such that

$$\begin{cases} \nabla f(x^*) + \sum_{j=1}^{m} \lambda^{*,j} \nabla g_j(x^*) = 0, \\ g_j(x^*) \le 0, \quad j = 1, \ldots, m, \\ \lambda^{*,j} g_j(x^*) = 0 \text{ and } \lambda^{*,j} \ge 0, \quad j = 1, \ldots, m. \end{cases} \tag{1}$$

It is well known that, under our assumptions, a necessary condition for optimality of a point $x^* \in X$ is that it be a KKT point.

Note that, with $x \in X$, $QP(x, H, \eta)$ is always consistent: $(0, 0)$ satisfies the constraints. Indeed, $QP(x, H, \eta)$ always has a unique solution (d, γ) (see Lemma 1 below) which, by convexity, is its unique KKT point; i.e. there exist multipliers μ and λ^j, $j = 1, \ldots, m$, which, together with (d, γ), satisfy

$$\begin{cases} \begin{bmatrix} Hd \\ 1 \end{bmatrix} + \mu \begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix} + \sum_{j=1}^{m} \lambda^j \begin{bmatrix} \nabla g_j(x) \\ -\eta \end{bmatrix} = 0, \\ \langle \nabla f(x), d \rangle \le \gamma, \\ g_j(x) + \langle \nabla g_j(x), d \rangle \le \gamma \cdot \eta, \quad j = 1, \ldots, m, \\ \mu \left(\langle \nabla f(x), d \rangle - \gamma \right) = 0 \text{ and } \mu \ge 0, \\ \lambda^j \left(g_j(x) + \langle \nabla g_j(x), d \rangle - \gamma \cdot \eta \right) = 0 \text{ and } \lambda^j \ge 0, \quad j = 1, \ldots, m. \end{cases} \tag{2}$$

A simple consequence of the first equation in (2), which will be used throughout our analysis, is an affine relationship amongst the multipliers, namely

$$\mu + \eta \cdot \sum_{j=1}^{m} \lambda^j = 1. \tag{3}$$

Parameter η will be assigned a new value at each iteration, η_k at iteration k, to ensure that $d(x_k, H_k, \eta_k)$ has the necessary properties. Strict positivity of η_k is sufficient to guarantee that Properties P1 to P3 are satisfied. As it turns out however, this is not enough to ensure that, away from a solution, there is adequate tilting into the feasible set. For this, we will force η_k to be bounded away from zero away from KKT points of (P). Finally, P4 requires that η_k tend to zero sufficiently fast as $d^0(x_k, H_k)$ tends to zero, i.e., as a solution is approached. In [14], a similar effect is achieved by first computing $d^0(x_k, H_k)$ but, of course, we want to avoid that here.

Given an estimate I_k^{E} of the active set $I(x_k)$, we can compute an estimate $d^{\mathrm{E}}(x_k, H_k, I_k^{\mathrm{E}})$ of $d^0(x_k, H_k)$ by solving the *equality* constrained QP

$$\begin{array}{ll} \min & \frac{1}{2}\langle d^{\mathrm{E}}, H_k d^{\mathrm{E}}\rangle + \langle \nabla f(x_k), d^{\mathrm{E}}\rangle \\ \text{s.t.} & g_j(x_k) + \langle \nabla g_j(x_k), d^{\mathrm{E}}\rangle = 0, \quad j \in I_k^{\mathrm{E}}, \end{array} \qquad LS^{\mathrm{E}}(x_k, H_k, I_k^{\mathrm{E}})$$

which is equivalent (after a change of variables) to solving a linear least squares problem. Let I_k be the set of active constraints, not including the "objective descent" constraint $\langle \nabla f(x_k), d_k\rangle \le \gamma_k$, for $QP(x_k, H_k, \eta_k)$, i.e.

$$I_k \triangleq \{\ j \mid g_j(x_k) + \langle \nabla g_j(x_k), d_k\rangle = \gamma_k \cdot \eta_k\ \}.$$

We will show in Section 3 that $d^{\mathrm{E}}(x_k, H_k, I_{k-1}) = d^0(x_k, H_k)$ for all k sufficiently large. Furthermore, we will prove that, when d_k is small, choosing

$$\eta_k \propto \|d^{\mathrm{E}}(x_k, H_k, I_{k-1})\|^2$$

is sufficient to guarantee global and local superlinear convergence. Proper choice of the proportionality constant (C_k in the algorithm statement below), while not important in the convergence analysis, is critical for satisfactory numerical performance. This will be discussed in Section 4.

In [15], given x, H, and a feasible descent direction d, the Maratos correction d^{C} (denoted $\tilde{d}$ in [15]) is taken as the solution of the QP

$$\begin{array}{ll} \min & \frac{1}{2}\langle d + d^{\mathrm{C}}, H(d + d^{\mathrm{C}})\rangle + \langle \nabla f(x), d + d^{\mathrm{C}}\rangle \\ \text{s.t.} & g_j(x + d) + \langle \nabla g_j(x), d + d^{\mathrm{C}}\rangle \le -\|d\|^\tau, \quad j = 1, \ldots, m, \end{array} \qquad QP^{\mathrm{C}}(x, d, H)$$

if it exists and has norm less than $\min\{\|d\|, C\}$, where τ is a given scalar satisfying $2 \le \tau \le 3$ and C a given large scalar. Otherwise, d^{C} is set to zero. (Indeed, a large d^{C} is meangingless and may jeopardize global convergence.) In Section 1, it was mentioned that a linear least squares problem could be used instead of a QP to compute a version of the Maratos correction d^{C} with the same asymptotic convergence properties. Given that our goal is to reduce the computational cost per iteration, it makes sense to use such an approach here. Thus, at iteration k, we take the correction d_k^{C} to be the solution $d^{\mathrm{C}}(x_k, d_k, H_k, I_k)$, if it exists and is not too large (specifically, if its norm is no larger than that of d_k), of the equality-constrained QP (equivalent to a least squares problem after a change of variables)

$$\begin{array}{ll} \min & \langle d_k + d^{\mathrm{C}}, H_k(d_k + d^{\mathrm{C}})\rangle + \langle \nabla f(x_k), d_k + d^{\mathrm{C}}\rangle \\ \text{s.t.} & g_j(x_k + d_k) + \langle \nabla g_j(x_k), d^{\mathrm{C}}\rangle = -\|d_k\|^\tau, \quad \forall j \in I_k, \end{array} \qquad LS^{\mathrm{C}}(x_k, d_k, H_k, I_k)$$

where $\tau \in (2,3)$, a direct extension of an alternative considered in [14]. In making use of the best available metric, such an objective, as compared to the pure least squares objective $\|d^{\mathrm{C}}\|^2$, should yield a somewhat better iterate without significantly increasing computational requirements (or affecting the convergence analysis). Another advantage of using metric H_k is that, asymptotically, the matrix underlying $LS^{\mathrm{C}}(x_k, d_k, H_k, I_k)$ will be the same as that underlying $LS^{\mathrm{E}}(x_k, H_k, I_{k-1})$, resulting in computational savings. In the case that $LS^{\mathrm{C}}(x_k, d_k, H_k, I_k)$ is inconsistent, or the computed solution d_k^{C} is too large, we will simply set d_k^{C} to zero.

The proposed algorithm is as follows. Parameters α, β are used in the Armijo-like search, τ is the "bending" exponent in LS^{C}, and ϵ_ℓ, $\underline{C}$, $\overline{C}$, and $\overline{D}$ are used in the update rule for η_k.

Algorithm FSQP$'$

Parameters: $\alpha \in (0, \frac{1}{2})$, $\beta \in (0,1)$, $\tau \in (2,3)$, $\epsilon_\ell > 0$, $0 < \underline{C} \leq \overline{C}$, $\bar{D} > 0$.

Data: $x_0 \in X$, H_0 positive definite, $\eta_0 > 0$.

Step 0 - Initialization. **set** $k \leftarrow 0$.

Step 1 - Computation of search arc.

(i). **compute** $(d_k, \gamma_k) = (d(x_k, H_k, \eta_k), \gamma(x_k, H_k, \eta_k))$, the active set I_k, and associated multipliers $\mu_k \in \mathbb{R}$, $\lambda_k \in \mathbb{R}^m$.
if $(d_k = 0)$ **then stop**.

(ii). **compute** $d_k^{\mathrm{C}} = d^{\mathrm{C}}(x_k, d_k, H_k, I_k)$ if it exists and satisfies $\|d_k^{\mathrm{C}}\| \leq \|d_k\|$. Otherwise, **set** $d_k^{\mathrm{C}} = 0$.

Step 2 - Arc search. **compute** t_k, the first value of t in the sequence $\{1, \beta, \beta^2, \dots\}$ that satisfies

$$f(x_k + td_k + t^2 d_k^{\mathrm{C}}) \leq f(x_k) + \alpha t \langle \nabla f(x_k), d_k \rangle,$$
$$g_j(x_k + td_k + t^2 d_k^{\mathrm{C}}) \leq 0, \quad j = 1, \dots, m.$$

Step 3 - Updates.

(i). **set** $x_{k+1} \leftarrow x_k + t_k d_k + t_k^2 d_k^{\mathrm{C}}$.

(ii). **compute** H_{k+1}, a new symmetric positive definite estimate to the Hessian of the Lagrangian.

(iii). **select** $C_{k+1} \in [\underline{C}, \overline{C}]$.

- $*$ **if** $(\|d_k\| < \epsilon_\ell)$ **then** if $LS^{\mathrm{E}}(x_{k+1}, H_{k+1}, I_k)$ has a unique solution and unique associated multipiers, **compute** $d^{\mathrm{E}}_{k+1} = d^{\mathrm{E}}(x_{k+1}, H_{k+1}, I_k)$, and the associated multipliers $\lambda^{\mathrm{E}}_{k+1} \in \mathbb{R}^{|I_k|}$. In such case,
  - $\cdot$ **if** $(\|d^{\mathrm{E}}_{k+1}\| \leq \bar{D}$ and $\lambda^{\mathrm{E}}_{k+1} \geq 0)$ **then set**

$$\eta_{k+1} \leftarrow C_{k+1} \cdot \|d^{\mathrm{E}}_{k+1}\|^2.$$

  - $\cdot$ **else set** $\eta_{k+1} \leftarrow C_{k+1} \cdot \|d_k\|^2$.
- $*$ **else set** $\eta_{k+1} \leftarrow C_{k+1} \cdot \epsilon_\ell^2$.

(iv). **set** $k \leftarrow k + 1$ and **goto** *Step 1.*

3 Convergence Analysis

Much of our analysis, especially the local analysis, will be devoted to establishing the relationship between $d(x, H, \eta)$ and the SQP direction $d^0(x, H)$. Given x in X and H symmetric positive definite, d^0 is a KKT point for $QP^0(x, H)$ (thus its unique solution $d^0(x, H)$) if and only if there exists a multiplier vector λ^0 such that

$$\begin{cases} Hd^0 + \nabla f(x) + \sum_{j=1}^{m} \lambda^{0,j} \nabla g_j(x) = 0, \\ g_j(x) + \langle \nabla g_j(x), d^0 \rangle \leq 0, \quad j = 1, \ldots, m, \\ \lambda^{0,j} \cdot \big(g_j(x) + \langle \nabla g_j(x), d^0 \rangle\big) = 0 \text{ and } \lambda^{0,j} \geq 0, \quad j = 1, \ldots, m. \end{cases} \tag{4}$$

Further, given $I \subset \{1, \ldots, m\}$, an estimate d^{E} is a KKT point for $LS^{\mathrm{E}}(x, H, I)$ (thus its unique solution $d^{\mathrm{E}}(x, H, I)$) if and only if there exists a multiplier vector λ^{E} such that

$$\begin{cases} Hd^{\mathrm{E}} + \nabla f(x) + \sum_{j \in I} \lambda^{\mathrm{E},j} \nabla g_j(x) = 0, \\ g_j(x) + \langle \nabla g_j(x), d^{\mathrm{E}} \rangle = 0, \quad j \in I. \end{cases} \tag{5}$$

Note that the components of λ^{E} for $j \notin I$ play no role in the optimality conditions.

3.1 Global Convergence

In this section we establish that, under mild assumptions, **FSQP′** generates a sequence of iterates $\{x_k\}$ with the property that all accumulation points are KKT points for (P). We begin by establishing some properties of the tilted SQP search direction $d(x, H, \eta)$.

Lemma 1. *Suppose Assumptions 1 through 3 hold. Then, given H symmetric positive definite, $x \in X$, and $\eta \geq 0$, $d(x, H, \eta)$ is well-defined and $(d(x, H, \eta), \gamma(x, H, \eta))$ is the unique KKT point of $QP(x, H, \eta)$. Furthermore, $d(x, H, \eta)$ is bounded over compact subsets of $X \times \mathcal{P} \times R^+$, where $\mathcal{P}$ is the set of symmetric positive definite $n \times n$ matrices and R^+ the set of nonnegative real numbers.*

Proof. First note that the feasible set for $QP(x, H, \eta)$ is non-empty, since $(d, \gamma) = (0, 0)$ is always feasible. Now consider the cases $\eta = 0$ and $\eta > 0$ separately. From (2) and (4), it is clear that, if $\eta = 0$, (d, γ) is a solution to $QP(x, H, 0)$ if and only if d is a solution of $QP^0(x, H)$ and $\gamma = \langle \nabla f(x), d \rangle$. It is well known that, under our assumptions, $d^0(x, H)$ is well-defined, unique, and continuous. The claims follow. Suppose now that $\eta > 0$. In that case, (d, γ) is a solution of $QP(x, H, \eta)$ if and only if d solves the unconstrained problem

$$\min \frac{1}{2}\langle d, Hd \rangle + \max\left\{ \langle \nabla f(x), d \rangle,\ \frac{1}{\eta} \cdot \max_{j=1,\dots,m} \{g_j(x) + \langle \nabla g_j(x), d \rangle\} \right\}. \quad (6)$$

and

$$\gamma = \max\left\{ \langle \nabla f(x), d \rangle,\ \frac{1}{\eta} \cdot \max_{j=1,\dots,m} \{g_j(x) + \langle \nabla g_j(x), d \rangle\} \right\}.$$

Since the function being minimized in (6) is strictly convex and radially unbounded, it follows that $(d(x, H, \eta), \gamma(x, H, \eta))$ is well-defined and unique as a global minimizer for the convex problem $QP(x, H, \eta)$, and thus unique as a KKT point for that problem. Boundedness of $d(x, H, \eta)$ over compact subsets of $X \times \mathcal{P} \times R^+$ follows from the first equation in (2), our regularity assumptions, and (3), which shows (since $\eta > 0$) that the multipliers are bounded. □

Lemma 2. *Suppose Assumptions 1 through 3 hold. Then, given H symmetric positive definite and $\eta \geq 0$*

(i). $\gamma(x, H, \eta) \leq 0$ *for all* $x \in X$. *Moreover,* $\gamma(x, H, \eta) = 0$ *if and only if* $d(x, H, \eta) = 0$.

(ii). $d(x, H, \eta) = 0$ if and only if x is a KKT point for (P). Moreover, if either (thus both) of these conditions holds, then the multipliers λ and μ for $QP(x, H, \eta)$ and λ^* for (P) are related by $\mu = (1 + \eta \sum_j \lambda^{*,j})^{-1}$ and $\lambda = \mu\lambda^*$.

Proof. To prove (i), note that since $(d, \gamma) = (0, 0)$ is always feasible for $QP(x, H, \eta)$, the optimal value of the QP is non-positive. Further, since $H > 0$, the quadratic term in the objective is non-negative, which implies $\gamma(x, H, \eta) \leq 0$. Now suppose that $d(x, H, \eta) = 0$, then feasibility of the first QP constraint implies that $\gamma(x, H, \eta) = 0$. Finally, suppose that $\gamma(x, H, \eta) = 0$. Since $x \in X$, $H > 0$, and $\eta \geq 0$, it is clear that $d = 0$ is feasible and achieves the minimum value of the objective. Thus, uniqueness gives $d(x, H, \eta) = 0$ and part (i) is proved.

Suppose now that $d(x, H, \eta) = 0$. Then $\gamma(x, H, \eta) = 0$ and by (2) there exist a multiplier vector λ and a scalar multiplier $\mu \geq 0$ such that

$$\begin{cases} \mu \nabla f(x) + \sum_{j=1}^{m} \lambda^j \nabla g_j(x) = 0, \\ g_j(x) \leq 0, \quad \forall j = 1, \ldots, m, \\ \lambda^j g_j(x) = 0 \text{ and } \lambda^j \geq 0, \quad \forall j = 1, \ldots, m. \end{cases} \tag{7}$$

We begin by showing that $\mu > 0$. Proceeding by contradiction, suppose $\mu = 0$, then by (3) we have

$$\sum_{j=1}^{m} \lambda^j > 0. \tag{8}$$

Note that,

$$\begin{aligned} I &\triangleq \{\, j \mid g_j(x) + \langle \nabla g_j(x), d(x, H, \eta) \rangle = \gamma(x, H, \eta) \cdot \eta \,\} \\ &= \{\, j \mid g_j(x) = 0 \,\} = I(x). \end{aligned}$$

Thus, by the complementary slackness condition of (2) and the optimality conditions (7),

$$0 = \sum_{j=1}^{m} \lambda^j \nabla g_j(x) = \sum_{j \in I(x)} \lambda^j \nabla g_j(x).$$

By Assumption 3, this sum vanishes only if $\lambda^j = 0$ for all $j \in I(x)$, contradicting (8). Thus $\mu > 0$. It is now immediate that x is a KKT point for (P) with multipliers $\lambda^{*,j} = \lambda^j / \mu$, $j = 1, \ldots, m$.

Finally, to prove the necessity portion of part (ii) note that if x is a KKT point for (P), then (1) shows that $(d, \gamma) = (0, 0)$ is a KKT point for $QP(x, H, \eta)$, with $\mu = (1 + \eta \sum_j \lambda^{*,j})^{-1}$ and $\lambda^j = \lambda^{*,j}(1 + \eta \sum_j \lambda^{*,j})^{-1}$, $j = 1, \ldots, m$. Uniqueness of such points (Lemma 1) yields the result. □

The next two lemmas establish that the line search in *Step 2* of Algorithm **FSQP′** is well defined.

Lemma 3. *Suppose Assumptions 1 through 3 hold. Suppose $x \in X$ is not a KKT point for (P), H is symmetric positive definite and $\eta > 0$. Then*

(i). $\langle \nabla f(x), d(x, H, \eta) \rangle < 0$, *and*

(ii). $\langle \nabla g_j(x), d(x, H, \eta) \rangle < 0$, *for all* $j \in I(x)$.

Proof. Both follow immediately from Lemma 2 and the fact that $d(x, H, \eta)$ and $\gamma(x, H, \eta)$ must satisfy the constraints in $QP(x, H, \eta)$. □

Lemma 4. *Suppose Assumptions 1 through 3 hold. Then, if $\eta_k = 0$, x_k is a KKT point for (P) and the algorithm will stop in Step 1(i) at iteration k. On the other hand, whenever the algorithm does not stop in Step 1(i), the line search is well defined, i.e. Step 2 yields a step t_k equal to β^{j_k} for some finite j_k.*

Proof. Suppose that $\eta_k = 0$. Then $k > 0$ and, by *Step 3(iii)*, either $d_k^{\mathrm{E}} = 0$ with $\lambda_k^{\mathrm{E}} \geq 0$, or $d_{k-1} = 0$. The latter case cannot hold, as the stopping criterion in *Step 1(i)* would have stopped the algorithm at iteration $k - 1$. On the other hand, if $d_k^{\mathrm{E}} = 0$ with $\lambda_k^{\mathrm{E}} \geq 0$, then in view of the optimality conditions (5), and the fact that x_k is always feasible for (P), we see that x_k is a KKT point for (P) with multipliers

$$\begin{cases} \lambda_k^{\mathrm{E},j}, & j \in I_{k-1}, \\ 0, & \text{otherwise}. \end{cases}$$

Thus, by Lemma 2, $d_k = 0$ and the algorithm will stop in *Step 1(i)*. The first claim is thus proved. Also, we have established that $\eta_k > 0$ whenever *Step 2* is reached. The second claim now follows immediately from Lemma 3 and Assumption 2. □

The previous lemmas imply that the algorithm is well-defined. In addition, Lemma 2 shows that if Algorithm **FSQP′** generates a finite sequence terminating at the point x_N, then x_N is a KKT point for the problem (P).

We now concentrate on the case in which an infinite sequence $\{x_k\}$ is generated, i.e. the algorithm never satisfies the termination condition in *Step 1(i)*. Note that, in view of Lemma 4, we may assume throughout that

$$\eta_k > 0, \quad \forall k. \tag{9}$$

Before proceeding, we make an assumption concerning the estimates H_k of the Hessian of the Lagrangian.

Assumption 4: There exist positive constants σ_1 and σ_2 such that, for all k,

$$\sigma_1 \|d\|^2 \leq \langle d, H_k d \rangle \leq \sigma_2 \|d\|^2, \quad \forall d \in \mathbb{R}^n.$$

Lemma 5. *Suppose Assumptions 1 through 4 hold. Then the sequence $\{\eta_k\}$ generated by Algorithm* **FSQP′** *is bounded. Further, the sequence $\{d_k\}$ is bounded on subsequences on which $\{x_k\}$ is bounded.*

Proof. The first claim follows from the update rule in *Step 3(iii)* of Algorithm **FSQP′** . The second claim then follows from Lemma 1 and Assumption 4. □

Given an infinite index set $\mathcal{K}$, we will use the notation

$$x_k \xrightarrow{k \in \mathcal{K}} x^*$$

to mean

$$x_k \to x^* \text{ as } k \to \infty, \ k \in \mathcal{K}.$$

Lemma 6. *Suppose Assumptions 1 through 3 hold. Suppose $\mathcal{K}$ is an infinite index set such that $x_k \xrightarrow{k \in \mathcal{K}} x^* \in X$, $\{\eta_k\}$ is bounded on $\mathcal{K}$, and $d_k \xrightarrow{k \in \mathcal{K}} 0$. Then $I_k \subseteq I(x^*)$, for all $k \in \mathcal{K}$, k sufficiently large and the QP multiplier sequences $\{\mu_k\}$ and $\{\lambda_k\}$ are bounded on $\mathcal{K}$. Further, given any accumulation point $\eta^* \geq 0$ of $\{\eta_k\}_{k \in \mathcal{K}}$, $(0,0)$ is the unique solution of $QP(x^*, H^*, \eta^*)$.*

Proof. In view of Assumption 2 $\{\nabla f(x_k)\}_{k \in \mathcal{K}}$ must be bounded. Lemma 2(i) and the first constraint in $QP(x_k, H_k, \eta_k)$ give

$$\langle \nabla f(x_k), d_k \rangle \leq \gamma_k \leq 0, \quad \forall k \in \mathcal{K}.$$

Thus, $\gamma_k \xrightarrow{k \in \mathcal{K}} 0$. To prove the first claim, let $j' \notin I(x^*)$. There exists $\delta_{j'} > 0$ such that $g_{j'}(x_k) \leq -\delta_{j'} < 0$, for all $k \in \mathcal{K}$, k sufficiently large. In view of

Assumption 2, and since $d_k \xrightarrow{k\in\mathcal{K}} 0$, $\gamma_k \xrightarrow{k\in\mathcal{K}} 0$, and $\{\eta_k\}$ is bounded on $\mathcal{K}$, it is clear that

$$g_{j'}(x_k) + \langle \nabla g_{j'}(x_k), d_k\rangle - \gamma_k \cdot \eta_k \leq -\frac{\delta_{j'}}{2} < 0,$$

i.e. $j' \notin I_k$, for all $k \in \mathcal{K}$, k sufficiently large, proving the first claim.

Boundedness of $\{\mu_k\}_{k\in\mathcal{K}}$ follows from non-negativity and (3). To prove that of $\{\lambda_k\}_{k\in\mathcal{K}}$, using complementary slackness and the first equation in (2), write

$$H_k d_k + \mu_k \nabla f(x_k) + \sum_{j\in I(x^*)} \lambda_k^j \nabla g_j(x_k) = 0. \tag{10}$$

Proceeding by contradiction, suppose that $\{\lambda_k\}_{k\in\mathcal{K}}$ is unbounded. Without loss of generality, assume that $\|\lambda_k\|_\infty > 0$, for all $k \in \mathcal{K}$ and define for all $k \in \mathcal{K}$

$$\nu_k^j \triangleq \frac{\lambda_k^j}{\|\lambda_k\|_\infty} \in [0,1].$$

Note that, for all $k \in \mathcal{K}$, $\|\nu_k\|_\infty = 1$. Dividing (10) by $\|\lambda_k\|_\infty$ and taking limits on an appropriate subsequence of $\mathcal{K}$, it follows from Assumptions 2 and 4 and boundedness of $\{\mu_k\}$ that

$$\sum_{j\in I(x^*)} \nu^{*,j} \nabla g_j(x^*) = 0,$$

for some $\nu^{*,j}$, $j \in I(x^*)$, where $\|\nu^*\|_\infty = 1$. As this contradicts Assumption 3, it is established that $\{\lambda_k\}_{k\in\mathcal{K}}$ is bounded.

To complete the proof, let $\mathcal{K}' \subseteq \mathcal{K}$ be an infinite index set such that $\eta_k \xrightarrow{k\in\mathcal{K}'} \eta^*$ and assume without loss of generality that $H_k \xrightarrow{k\in\mathcal{K}'} H^*$, $\mu_k \xrightarrow{k\in\mathcal{K}'} \mu^*$ and $\lambda_k \xrightarrow{k\in\mathcal{K}'} \lambda^*$. Taking limits in the optimality conditions (2) shows that, indeed, $(d,\gamma) = (0,0)$ is a KKT point for $QP(x^*, H^*, \eta^*)$ with multipliers μ^* and λ^*. Finally, uniqueness of such points (Lemma 1) proves the result. $\square$

Lemma 7. *Suppose Assumptions 1 through 4 hold. Then, if $\mathcal{K}$ is an infinite index set such that $d_k \xrightarrow{k\in\mathcal{K}} 0$, all accumulation points of $\{x_k\}_{k\in\mathcal{K}}$ are KKT points for (P).*

Proof. Suppose $\mathcal{K}' \subseteq \mathcal{K}$ is an infinite index set on which $x_k \xrightarrow{k\in\mathcal{K}'} x^* \in X$. In view of Assumption 4 and Lemma 5, assume, without loss of generality that $H_k \xrightarrow{k\in\mathcal{K}'} H^*$, a positive definite matrix, and $\eta_k \xrightarrow{k\in\mathcal{K}'} \eta^* \geq 0$. In view of Lemma 6, $(0,0)$ is the unique solution of $QP(x^*, H^*, \eta^*)$. It follows from Lemma 2 that x^* is a KKT point for (P). $\square$

We now state and prove the main result of this subsection.

Theorem 1. *Under Assumptions 1 through 4, Algorithm* **FSQP′** *generates a sequence* $\{x_k\}$ *for which all accumulation points are KKT points for* (P).

Proof. Suppose $\mathcal{K}$ is an infinite index set such that $x_k \xrightarrow{k\in\mathcal{K}} x^*$. In view of Lemma 5 and Assumption 4, we may assume without loss of generality that $d_k \xrightarrow{k\in\mathcal{K}} d^*$, $\eta_k \xrightarrow{k\in\mathcal{K}} \eta^* \geq 0$, and $H_k \xrightarrow{k\in\mathcal{K}} H^* > 0$. The cases $\eta^* = 0$ and $\eta^* > 0$ are considered separately.

Suppose first that $\eta^* = 0$. Then, by *Step 3(iii)*, there exists an infinite index set $\mathcal{K}' \subseteq \mathcal{K}$ such that either $d_k^{\mathrm{E}} \xrightarrow{k\in\mathcal{K}'} 0$ with $\lambda_k^{\mathrm{E}} \geq 0$, for all $k \in \mathcal{K}'$, or $d_{k-1} \xrightarrow{k\in\mathcal{K}'} 0$. If the latter case holds, it is then clear that $x_{k-1} \xrightarrow{k\in\mathcal{K}'} x^*$, since $\|x_k - x_{k-1}\| \leq 2\|d_{k-1}\| \xrightarrow{k\in\mathcal{K}'} 0$. Thus, by Lemma 7, x^* is a KKT point for (P). Now suppose instead that $d_k^{\mathrm{E}} \xrightarrow{k\in\mathcal{K}'} 0$ with $\lambda_k^{\mathrm{E}} \geq 0$, for all $k \in \mathcal{K}'$. From the second set of equations in (5), one can easily see that $I_{k-1} \subseteq I(x^*)$ for all $k \in \mathcal{K}'$, k sufficiently large, and using an argument very similar to that used in Lemma 6, one can show that $\{\lambda_k^{\mathrm{E}}\}_{k\in\mathcal{K}'}$ is a bounded sequence. Thus, taking limits in (5) on an appropriate subsequence of $\mathcal{K}'$ shows that x^* is a KKT point for (P).

Now consider the case $\eta^* > 0$. We show that $d_k \xrightarrow{k\in\mathcal{K}} 0$. Proceeding by contradiction, without loss of generality suppose there exists $\underline{d} > 0$ such that $\|d_k\| \geq \underline{d}$ for all $k \in \mathcal{K}$. From non-positivity of the optimal value of the objective function in $QP(x_k, H_k, \eta_k)$ (since $(0, 0)$ is always feasible) and Assumption 4, we see that

$$\gamma_k \leq -\frac{1}{2}\sigma_1 \underline{d}^2 < 0, \quad \forall k \in \mathcal{K}.$$

Further, in view of (9) and since $\eta^* > 0$, there exists $\underline{\eta} > 0$ such that

$$\eta_k > \underline{\eta}, \quad \forall k \in \mathcal{K}.$$

From the constraints of $QP(x_k, H_k, \eta_k)$, it follows that

$$\langle \nabla f(x_k), d_k \rangle \leq -\frac{1}{2}\sigma_1 \underline{d}^2 < 0, \quad \forall k \in \mathcal{K},$$

and

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k \rangle \leq -\frac{1}{2}\sigma_1 \underline{d}^2 \underline{\eta} < 0, \quad \forall k \in \mathcal{K},$$

$j = 1, \ldots, m$. Hence, using Assumption 2, it is easily shown that there exists $\delta > 0$ such that for all $k \in \mathcal{K}$, k large enough,

$$\langle \nabla f(x_k), d_k \rangle \leq -\delta,$$

$$\langle \nabla g_j(x_k), d_k \rangle \leq -\delta, \quad \forall j \in I(x^*)$$

$$g_j(x_k) \leq -\delta, \quad \forall j \in \{1, \ldots, m\} \setminus I(x^*).$$

The rest of the contradiction argument establishing $d_k \xrightarrow{k \in \mathcal{K}} 0$ follows exactly the proof of Proposition 3.2 in [14]. Finally, it then follows from Lemma 7 that x^* is a KKT point for (P). □

3.2 Local Convergence

While the details are often quite different, overall the analysis in this section is inspired by and occasionally follows that of Panier and Tits in [14, 15]. The key result is Proposition 1 which states that, under appropriate assumptions, the arc search eventually accepts the full step of one. With this and the fact, to be established along the way, that titled direction d_k approaches the standard SQP direction sufficiently fast, superlinear convergence follows from a classical analysis of M.J.D. Powell's. As a first step, we strengthen the regularity assumptions.

Assumption 2$'$: The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, m$, are three times continuously differentiable.

A point x^* is said to satisfy the *second order sufficiency conditions with strict complementary slackness* for (P) if there exists a multiplier vector $\lambda^* \in \mathbb{R}^m$ such that

- The pair (x^*, λ^*) satisfies (1), i.e. x^* is a KKT point for (P),
- $\nabla^2_{xx} L(x^*, \lambda^*)$ is positive definite on the subspace

$$\{h \mid \langle \nabla g_j(x^*), h \rangle = 0, \ \forall j \in I(x^*)\},$$

- and $\lambda^{*,j} > 0$ for all $j \in I(x^*)$ (strict complementary slackness).

In order to guarantee that the entire sequence $\{x_k\}$ converges to a KKT point x^*, we make the following assumption. (Recall that we have already established, under weaker assumptions, that *every* accumulation point of $\{x_k\}$ is a KKT point for (P).)

Assumption 5: The sequence $\{x_k\}$ has an accumulation point x^* which satisfies the second order sufficiency conditions with strict complementary slackness.

It is well known that Assumption 5 guarantees that the entire sequence converges. For a proof see, e.g., Proposition 4.1 in [14].

Lemma 8. *Suppose Assumptions 1, 2', and 3 through 5 hold. Then the entire sequence generated by Algorithm* $\mathbf{FSQP'}$ *converges to a point* x^* *satisfying the second order sufficiency conditions with strict complementary slackness.*

From this point forward, λ^* will denote the (unique) multiplier vector associated with KKT point x^* for (P). It is readily checked that, for any symmetric positive definite H, $(0, \lambda^*)$ is the KKT pair for $QP^0(x^*, H)$.

As announced, as a first main step, we show that our sequence of tilted SQP directions approaches the true SQP direction sufficiently fast. (This is achieved in Lemmas 9 through 18.) In order to do so, define d_k^0 to be equal to $d^0(x_k, H_k)$, where x_k and H_k are as computed by Algorithm $\mathbf{FSQP'}$. Further, for each k, define λ_k^0 as a multiplier vector such that (d_k^0, λ_k^0) satisfy (4) and let $I_k^0 \triangleq \{ j \mid g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle = 0 \}$. The following lemma is proved in [15] (with reference to [14]) under identical assumptions.

Lemma 9. *Suppose Assumptions 1, 2', and 3 through 5 hold. Then*

(i) $d_k^0 \to 0$,

(ii) $\lambda_k^0 \to \lambda^*$.

(iii) For all k sufficiently large, the following two equalities hold

$$I_k^0 = \{ j \mid \lambda_k^{0,j} > 0 \} = I(x^*).$$

We next establish that the entire tilted SQP direction sequence converges to 0. In order to do so, we establish that $d(x, H, \eta)$ is continuous in a neighborhood of (x^*, H^*, η^*), for any $\eta^* \geq 0$ and H^* symmetric positive definite. Complicating the analysis is the fact that we have yet to establish that the sequence $\{\eta_k\}$ does, in fact, converge. Given $\eta^* \geq 0$, define the set

$$N^*(\eta^*) \triangleq \left\{ \begin{pmatrix} \nabla f(x^*) \\ -1 \end{pmatrix}, \begin{pmatrix} \nabla g_j(x^*) \\ -\eta^* \end{pmatrix}, j \in I(x^*) \right\}.$$

Lemma 10. *Suppose Assumptions 1, 2', and 3 through 5 hold. Then, given any* $\eta^* \geq 0$*, the set* $N^*(\eta^*)$ *is linearly independent.*

Proof. Let H^* be symmetric positive definite. Note that, in view of Lemma 2, $d(x^*, H^*, \eta^*) = 0$. Now suppose the claim does not hold, i.e. suppose there exists scalars λ^j, $j \in \{0\} \cup I(x^*)$, not all zero, such that

$$\lambda^0 \begin{pmatrix} \nabla f(x^*) \\ -1 \end{pmatrix} + \sum_{j \in I(x^*)} \lambda^j \begin{pmatrix} \nabla g_j(x^*) \\ -\eta^* \end{pmatrix} = 0. \tag{11}$$

In view of Assumption 3, $\lambda^0 \neq 0$ and the scalars λ^j are unique modulo a scaling factor. This uniqueness, the fact that $d(x^*, H^*, \eta^*) = 0$, and the first n scalar equations in the optimality conditions (2) imply that $\mu^* = 1$ and

$$\lambda^{*,j} = \begin{cases} \dfrac{\lambda^j}{\lambda^0} & j \in I(x^*) \\ 0 & \text{else,} \end{cases}$$

$j = 1, \ldots, m$ are KKT multipliers for $QP(x^*, H^*, \eta^*)$. Thus, in view of (3),

$$\eta^* \cdot \sum_{j \in I(x^*)} \frac{\lambda^j}{\lambda^0} = 0.$$

But this contradicts (11), which gives

$$\eta^* \cdot \sum_{j \in I(x^*)} \frac{\lambda^j}{\lambda^0} = -1,$$

hence $N^*(\eta^*)$ is linearly independent. $\square$

Lemma 11. *Suppose Assumptions 1, 2', and 3 through 5 hold. Let* $\eta^* \geq 0$ *be an accumulation point of* $\{\eta_k\}$*. Then, given any symmetric positive definite* H*,* $(d^*, \gamma^*) = (0, 0)$ *is the unique solution of* $QP(x^*, H, \eta^*)$ *and the second order sufficiency conditions hold, with strict complementary slackness.*

Proof. In view of Lemma 2, $QP(x^*, H, \eta^*)$ has $(d^*, \gamma^*) = (0, 0)$ as its unique solution. Define the Lagrangian function $\mathcal{L} : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}$ for $QP(x^*, H, \eta^*)$ as

$$\begin{aligned} \mathcal{L}(d, \gamma, \mu, \lambda) &= \frac{1}{2}\langle d, Hd\rangle + \gamma + \mu\left(\langle \nabla f(x^*), d\rangle - \gamma\right) \\ &\quad + \sum_{j=1}^{m} \lambda^j \left(g_j(x^*) + \langle \nabla g_j(x^*), d\rangle - \gamma\eta^*\right). \end{aligned}$$

Suppose $\hat{\mu} \in \mathbb{R}$ and $\hat{\lambda} \in \mathbb{R}^m$ are KKT multipliers such that (2) holds with $d = 0$, $\gamma = 0$, $\mu = \hat{\mu}$ and $\lambda = \hat{\lambda}$. Let $j = 0$ be the index for the first constraint in $QP(x^*, H, \eta^*)$, i.e. $\langle \nabla f(x^*), d \rangle \leq \gamma$. Note that since $(d^*, \gamma^*) = (0, 0)$, the active constraint index set I^* for $QP(x^*, H, \eta^*)$ is equal to $I(x^*) \cup \{0\}$. (Note that we define I^* as including 0, while I_k was defined as a subset of $\{1, \ldots, m\}$.) Thus the set of active constraint gradients for $QP(x^*, H, \eta^*)$ is $N^*(\eta^*)$.

Now consider the Hessian of the Lagrangian for $QP(x^*, H, \eta^*)$, i.e. the second derivative with respect to the first two variables (d, γ),

$$\nabla^2 \mathcal{L}(0, 0, \hat{\lambda}, \hat{\mu}) = \begin{bmatrix} H & 0 \\ 0 & 0 \end{bmatrix},$$

and given an arbitrary $h \in \mathbb{R}^{n+1}$, decompose it as $h = (y^T, \alpha)^T$, where $y \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$. Then clearly,

$$\langle h, \nabla^2 \mathcal{L}(0, 0, \hat{\lambda}, \hat{\mu}) h \rangle \geq 0, \quad \forall h$$

and for $h \neq 0$, $h^T \nabla^2 \mathcal{L}(0, 0, \hat{\lambda}, \hat{\mu}) h = y^T H y$ is zero if and only if $y = 0$ and $\alpha \neq 0$. Since for such h

$$\begin{pmatrix} \nabla f(x^*) \\ -1 \end{pmatrix}^T \begin{pmatrix} 0 \\ \alpha \end{pmatrix} = -\alpha \neq 0,$$

it then follows that $\nabla^2 \mathcal{L}(0, 0, \hat{\lambda}, \hat{\mu})$ is positive definite on $N^*(\eta^*)^\perp$, the tangent space to the active constraints for $QP(x^*, H, \eta^*)$ at $(0, 0)$. Thus, it is established that the second order sufficiency conditions hold.

Finally it follows from Lemma 2(ii) that $\hat{\mu} > 0$ and $\hat{\lambda} = \hat{\mu} \lambda^*$ which, together with Assumption 5, implies strict complementarity for $QP(x^*, H, \eta^*)$ at $(0, 0)$. $\square$

Lemma 12. *Suppose Assumptions 1, 2', and 3 through 5 hold. Then, if $\mathcal{K}$ is a subsequence on which $\{\eta_k\}$ converges, say to $\eta^* \geq 0$, then $\mu_k \xrightarrow{k \in \mathcal{K}} \hat{\mu} > 0$ and $\lambda_k \xrightarrow{k \in \mathcal{K}} \hat{\mu} \lambda^*$, where $\hat{\mu} = (1 + \eta^* \sum_j \lambda^{*,j})^{-1}$. Finally, $d_k \to 0$ and $\gamma_k \to 0$.*

Proof. First, proceed by contradiction to show that the first two claims hold and that, in addition,

$$(d_k, \gamma_k) \xrightarrow{k \in \mathcal{K}} (0, 0); \tag{12}$$

i.e., suppose that on some infinite index set $\mathcal{K}' \subseteq \mathcal{K}$ either μ_k is bounded away from $\hat{\mu}$, or λ_k is bounded away from $\hat{\mu} \lambda^*$, or (d_k, γ_k) is bounded away from zero. In view of Assumption 4, there is no loss of generality is assuming that $H_k \xrightarrow{k\in\mathcal{K}'} H^*$ for some symmetric positive definite H^*. In view of Lemmas 10 and 11, we may thus invoke a result due to Robinson (Theorem 2.1 in [21]) to conclude that, in view of Lemma 2(ii),

$$(d_k, \gamma_k) \xrightarrow{k\in\mathcal{K}'} (0,0), \quad \mu_k \xrightarrow{k\in\mathcal{K}'} \hat{\mu}, \quad \lambda_k \xrightarrow{k\in\mathcal{K}'} \hat{\mu}\lambda^*,$$

a contradiction. Hence the first two claims hold, as does (12). Next, proceeding again by contradiction, suppose that $d_k \not\to 0$. Then, since $\{H_k\}$ and $\{\eta_k\}$ are bounded, there exists an infinite index set $\mathcal{K}$ on which $\{H_k\}$ and $\{\eta_k\}$ converge and d_k is bounded away from zero. This contradicts (12). Thus $d_k \to 0$. It immediately follows from the first constraint in $QP(x_k, H_k, \eta_k)$ that $\gamma_k \to 0$. $\square$

Lemma 13. *Suppose Assumptions 1, 2', and 3 through 5 hold. Then, for all k sufficiently large, $I_k = I(x^*)$.*

Proof. Since $\{\eta_k\}$ is bounded and, in view of Lemma 12, $(d_k, \gamma_k) \to (0,0)$, Lemma 6 implies that $I_k \subseteq I(x^*)$, for all k sufficiently large. Now suppose it does not hold that $I_k = I(x^*)$ for all k sufficiently large. Thus, there exists $j' \in I(x^*)$ and an infinite index set $\mathcal{K}$ such that $j' \notin I_k$, for all $k \in \mathcal{K}$. Now, in view of Lemma 5, there exists an infinite index set $\mathcal{K}' \subseteq \mathcal{K}$ and $\eta^* \geq 0$ such that $\eta_k \xrightarrow{k\in\mathcal{K}'} \eta^*$. Since $j' \in I(x^*)$, Assumption 5 guarantees $\lambda^{*,j'} > 0$. Further, Lemma 12 shows that $\lambda_k^{j'} \xrightarrow{k\in\mathcal{K}'} \hat{\mu}\lambda^{*,j'} > 0$. Therefore, $\lambda_k^{j'} > 0$ for all k sufficiently large, $k \in \mathcal{K}'$, which, by complementary slackness, implies $j' \in I_k$ for all $k \in \mathcal{K}'$ large enough. Since $\mathcal{K}' \subseteq \mathcal{K}$, this is a contradiction, and the claim is proved. $\square$

Now define

$$\begin{aligned} R_k &\triangleq [\ \nabla g_j(x_k) \ : \ j \in I(x^*)\ ], \\ g_k &\triangleq [\ g_j(x_k) \ : \ j \in I(x^*)\ ]^T. \end{aligned}$$

and, given a vector $\lambda \in \mathbb{R}^m$, define the notation

$$\lambda^+ \triangleq [\ \lambda^j \ : \ j \in I(x^*)\ ]^T.$$

Note that, in view of Lemma 9(iii), for k large enough, the optimality conditions (4), yield

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix} \begin{pmatrix} d_k^0 \\ (\lambda_k^0)^+ \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) \\ g_k \end{pmatrix}. \tag{13}$$

The following well-known result will be used.

Lemma 14. *Suppose Assumptions 1, 2', and 3 through 5 hold. Then the matrix*

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix}$$

is invertible for all k large enough and its inverse remains bounded as $k \to \infty$.

Lemma 15. *Suppose Assumptions 1, 2', and 3 through 5 hold. For all k sufficiently large, d_k^{E} and λ_k^{E} are uniquely defined, and $d_k^{\mathrm{E}} = d_k^0$.*

Proof. In view of Lemma 13, the optimality conditions (5), and Lemma 14, for all k large enough, the estimate d_k^{E} and its corresponding multiplier vector λ_k^{E} are well defined as the unique solution of

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix} \begin{pmatrix} d_k^{\mathrm{E}} \\ (\lambda_k^{\mathrm{E}})^+ \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) \\ g_k \end{pmatrix}. \tag{14}$$

The claim then follows from (13). □

Lemma 16. *Suppose Assumptions 1, 2', and 3 through 5 hold. Then*

(*i*) *$\eta_k \to 0$,*

(*ii*) *$\mu_k \to 1$, and $\lambda_k \to \lambda^*$.*

(*iii*) *For all k sufficiently large, $I_k = \{ \ j \mid \lambda_k^j > 0 \ \}$.*

Proof. Claim (i) follows from *Step 3(iii)* of Algorithm **FSQP′**, since in view of Lemma 12, Lemma 15, and Lemma 9, $\{d_k\}$ and $\{d_k^{\mathrm{E}}\}$ both converge to 0. In view of (i), Lemma 12 establishes that $\mu_k \to 1$, and $\lambda_k \to \lambda^*$, hence claim (ii) is proved. Finally, claim (iii) follows from claim (ii), Lemma 13, and Assumption 5. □

We now focus our attention on establishing relationships between d_k, d_k^{C}, and the true SQP direction d_k^0.

Lemma 17. *Suppose Assumptions 1, 2', and 3 through 5 hold. Then*

(*i*) *$\eta_k = O(\|d_k^0\|^2)$,*

(*ii*) *$d_k = d_k^0 + O(\|d_k^0\|^2)$.*

(*iii*) *$\gamma_k = O(\|d_k^0\|)$.*

Proof. In view of Lemma 15, for all k sufficiently large, d_k^{E} and λ_k^{E} exist and are uniquely defined, and $d_k^{\mathrm{E}} = d_k^0$. Lemmas 12 and 9 ensure that *Step 3(iii)* of Algorithm **FSQP′** chooses $\eta_k = C_k \cdot \|d_k^{\mathrm{E}}\|^2$ for all k sufficiently large, thus (i) follows. It is clear from Lemma 13 and the optimality conditions (2) that d_k and λ_k satisfy

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix} \begin{pmatrix} d_k \\ \lambda_k^+ \end{pmatrix} = -\begin{pmatrix} \mu_k \cdot \nabla f(x_k) \\ g_k - \eta_k \cdot \gamma_k \cdot \mathbf{1}_{|I(x^*)|} \end{pmatrix}$$

$$= -\begin{pmatrix} \nabla f(x_k) \\ g_k \end{pmatrix} + \eta_k \cdot \begin{pmatrix} \left(\displaystyle\sum_{j \in I(x^*)} \lambda_k^j \right) \cdot \nabla f(x_k) \\ \gamma_k \cdot \mathbf{1}_{|I(x^*)|} \end{pmatrix}, \quad (15)$$

for all k sufficiently large, where $\mathbf{1}_{|I(x^*)|}$ is a vector of $|I(x^*)|$ ones. It thus follows from (13), Assumption 2, and Lemmas 12, 14 and 16 that

$$d_k = d_k^0 + O(\eta_k),$$

and in view of claim (i), claim (ii) follows. Finally, since (from the QP constraint and Lemma 2) $\langle \nabla f(x_k), d_k \rangle \leq \gamma_k < 0$, it is clear that $\gamma_k = O(\|d_k\|) = O(\|d_k^0\|)$. □

Lemma 18. *Suppose Assumptions 1, 2', and 3 through 5 hold. Then* $d_k^{\mathrm{C}} = O(\|d_k^0\|^2)$.

Proof. Let

$$c_k \triangleq [-g_j(x_k + d_k) - \|d_k\|^\tau \;:\; j \in I(x^*)]^T.$$

Expanding $g_j(\cdot)$, $j \in I(x^*)$, about x_k we see that, for some $\xi^j \in (0,1)$, $j \in I(x^*)$,

$$c_k = [\; \overbrace{-g_j(x_k) - \langle \nabla g_j(x_k), d_k \rangle}^{= -\eta_k \cdot \gamma_k} + \frac{1}{2} \langle d_k, \nabla^2 g_j(x_k + \xi^j d_k) d_k \rangle - \|d_k\|^\tau \;:\; j \in I(x^*) \;]^T.$$

Since $\tau > 2$, from Lemma 17 and Assumption 2' we conclude that $c_k = O(\|d_k^0\|^2)$. Now, for all k sufficiently large, in view of Lemma 13, d_k^{C} is well-defined and satisfies

$$g_j(x_k + d_k) + \langle \nabla g_j(x_k), d_k^{\mathrm{C}} \rangle = -\|d_k\|^\tau, \quad j \in I(x^*), \quad (16)$$

thus

$$R_k^T d_k^{\mathrm{C}} = c_k \tag{17}$$

Now, the first order KKT conditions for $LS^{\mathrm{C}}(x_k, d_k, H_k, I_k)$ tell us there exists a multiplier ${\lambda^{\mathrm{C}}}_k \in \mathbb{R}^{|I(x^*)|}$ such that

$$\begin{cases} H_k(d_k + d_k^{\mathrm{C}}) + \nabla f(x_k) + R_k {\lambda^{\mathrm{C}}}_k = 0, \\ R_k^T d_k^{\mathrm{C}} = c_k. \end{cases}$$

Also, from the optimality conditions (15) we have

$$H_k d_k + \nabla f(x_k) = q_k - R_k \lambda_k^+,$$

where

$$q_k \triangleq \eta_k \cdot \left(\sum_{j \in I(x^*)} \lambda_k^j \right) \cdot \nabla f(x_k).$$

In view of Lemma 17, $q_k = O(\|d_k^0\|^2)$. So, d_k^{C} and ${\lambda^{\mathrm{C}}}_k$ satisfy

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix} \begin{pmatrix} d_k^{\mathrm{C}} \\ {\lambda^{\mathrm{C}}}_k \end{pmatrix} = \begin{pmatrix} R_k \lambda_k^+ - q_k \\ c_k \end{pmatrix},$$

or equivalently, with $\lambda_k' = {\lambda^{\mathrm{C}}}_k - \lambda_k^+$,

$$\begin{bmatrix} H_k & R_k \\ R_k^T & 0 \end{bmatrix} \begin{pmatrix} d_k^{\mathrm{C}} \\ \lambda_k' \end{pmatrix} = \begin{pmatrix} -q_k \\ c_k \end{pmatrix} = O(\|d_k^0\|^2),$$

The result then follows from Lemma 14. □

In order to prove the key result that the full step of one is eventually accepted by the line search, we now assume that the matrices $\{H_k\}$ suitably approximate the Hessian of the Lagrangian at the solution. Define the projection

$$P_k \triangleq I - R_k (R_k^T R_k)^{-1} R_k^T.$$

Assumption 6:

$$\lim_{k \to \infty} \frac{\|P_k (H_k - \nabla_{xx}^2 L(x^*, \lambda^*)) P_k d_k\|}{\|d_k\|} = 0.$$

The following technical lemma will be used.

Lemma 19. *Suppose Assumptions 1, 2', and 3 through 5 hold. Then there exist constants* $\nu_1,\ \nu_2,\ \nu_3 > 0$ *such that*

(i) $\langle \nabla f(x_k), d_k \rangle \le -\nu_1 \|d_k^0\|^2$,

(ii) for all k *sufficiently large*

$$\sum_{j \in I(x^*)} \lambda_k^j g_j(x_k) \le -\nu_2 \|g_k\|,$$

(iii) $d_k = P_k d_k + d_k^1$, *where, for all* k *sufficiently large,*

$$\|d_k^1\| \le \nu_3 \|g_k\| + O(\|d_k^0\|^3).$$

Proof. To show part (i), note that in view of the first QP constraint, negativity of the optimal value of the QP objective, and Assumption 4,

$$\begin{aligned}
\langle \nabla f(x_k), d_k \rangle &\le \gamma_k \\
&\le -\tfrac{1}{2}\langle d_k, H_k d_k \rangle \\
&\le -\tfrac{\sigma_1}{2}\|d_k\|^2 = -\tfrac{\sigma_1}{2}\|d_k^0\|^2 + O(\|d_k^0\|^4).
\end{aligned}$$

The proof of part (ii) is identical to that of Lemma 4.4 in [14]. To show (iii), note that from (15) for all k sufficiently large, d_k satisfies

$$R_k^T d_k = -g_k - \gamma_k \eta_k \cdot \mathbf{1}_{|I(x^*)|}.$$

Thus, we can write $d_k = P_k d_k + d_k^1$, where

$$d_k^1 = -R_k (R_k^T R_k)^{-1} (g_k + \gamma_k \eta_k \cdot \mathbf{1}_{|I(x^*)|}).$$

The result follows from Assumption 3 and Lemma 17(i,iii). □

Proposition 1. *Suppose Assumptions 1, 2', and 3 through 6 hold. Then,* $t_k = 1$ *for all* k *sufficiently large.*

Proof. Following [14], consider an expansion of $g_j(\cdot)$ about $x_k + d_k$ for $j \in I(x^*)$, for all k sufficiently large,

$$\begin{aligned}
g_j(x_k + d_k + d_k^{\mathrm{C}}) &= g_j(x_k + d_k) + \langle \nabla g_j(x_k + d_k), d_k^{\mathrm{C}} \rangle + O(\|d_k^0\|^4) \\
&= g_j(x_k + d_k) + \langle \nabla g_j(x_k), d_k^{\mathrm{C}} \rangle + O(\|d_k^0\|^3) \\
&= -\|d_k\|^\tau + O(\|d_k^0\|^3) \\
&= -\|d_k^0\|^\tau + O(\|d_k^0\|^3),
\end{aligned}$$

where we have used Assumption 2', Lemmas 17 and 18, boundedness of all sequences, and (16). As $\tau < 3$, it follows that $g_j(x_k+d_k+d_k^{\mathrm{C}}) \leq 0$, $j \in I(x^*)$, for all k sufficiently large. The same result trivially holds for $j \notin I(x^*)$. Thus, for k large enough, the full step of one satisfies the feasibility condition in the arc search test. It remains to show that the "sufficient decrease" condition is satisfied as well.

First, in view of Assumption $2'$ and Lemmas 17 and 18,

$$\begin{aligned} f(x_k + d_k + d_k^{\mathrm{C}}) &= f(x_k) + \langle \nabla f(x_k), d_k \rangle + \langle \nabla f(x_k), d_k^{\mathrm{C}} \rangle \\ &\quad + \tfrac{1}{2}\langle d_k, \nabla^2 f(x_k) d_k \rangle + O(\|d_k^0\|^3). \end{aligned} \tag{18}$$

From the top equation in optimality conditions (2), equation (3), Lemma 17(i), and boundedness of all sequences, we obtain

$$H_k d_k + \nabla f(x_k) + \sum_{j=1}^{m} \lambda_k^j \nabla g_j(x_k) = O(\|d_k^0\|^2). \tag{19}$$

The last line in (2) and Lemma 17(i,iii) yield

$$\lambda_k^j \langle \nabla g_j(x_k), d_k \rangle = -\lambda_k^j g_j(x_k) + O(\|d_k^0\|^3). \tag{20}$$

Taking the inner product of (19) with d_k, then adding and subtracting the quantity $\sum_j \lambda_k^j \langle \nabla g_j(x_k), d_k \rangle$, using (20), and finally multiplying the result by $\frac{1}{2}$ gives

$$\begin{aligned} \tfrac{1}{2}\langle \nabla f(x_k), d_k \rangle &= -\frac{1}{2}\langle d_k, H_k d_k \rangle - \sum_{j=1}^{m} \lambda_k^j \langle \nabla g_j(x_k), d_k \rangle \\ &\quad - \frac{1}{2}\sum_{j=1}^{m} \lambda_k^j g_j(x_k) + O(\|d_k^0\|^3). \end{aligned} \tag{21}$$

Further, Lemmas 17 and 18 and equation (19) give

$$\langle \nabla f(x_k), d_k^{\mathrm{C}} \rangle = -\sum_{j=1}^{m} \lambda_k^j \langle \nabla g_j(x_k), d_k^{\mathrm{C}} \rangle + O(\|d_k^0\|^3). \tag{22}$$

Combining (18), (21), and (22), and using the fact that, for k large enough, $\lambda_k^j = 0$ for all $j \notin I(x^*)$ (Lemma 9(iii)), we obtain

$$f(x_k + d_k + d_k^{\mathrm{C}}) - f(x_k) =$$

$$\frac{1}{2}\langle \nabla f(x_k), d_k\rangle - \frac{1}{2}\langle d_k, H_k d_k\rangle - \frac{1}{2}\sum_{j\in I(x^*)} \lambda_k^j g_j(x_k)$$

$$-\sum_{j\in I(x^*)} \lambda_k^j \langle \nabla g_j(x_k), d_k\rangle - \sum_{j\in I(x^*)} \lambda_k^j \langle \nabla g_j(x_k), d_k^{\mathrm{C}}\rangle$$

$$+\frac{1}{2}\langle d_k, \nabla^2 f(x_k) d_k\rangle + O(\|d_k^0\|^3).$$

With this in hand, arguments identical to those used following equation (4.9) in [14] show that

$$f(x_k + d_k + d_k^{\mathrm{C}}) - f(x_k) - \alpha\langle \nabla f(x_k), d_k\rangle < 0$$

for all k sufficiently large. Thus the "sufficient decrease" condition is satisfied. □

A consequence of Lemmas 17, 18, and Proposition 1 is that the algorithm generates a convergent sequence of iterates satisfying

$$x_{k+1} - x_k = d_k^0 + O(\|d_k^0\|^2).$$

Two-step superlinear convergence follows.

Theorem 2. *Suppose Assumptions 1, 2', and 3 through 6 hold. Then Algorithm* **FSQP′** *generates a sequence* $\{x_k\}$ *which converges 2-step superlinearly to* x^*, *i.e.*

$$\lim_{k\to\infty} \frac{\|x_{k+2} - x^*\|}{\|x_k - x^*\|} = 0.$$

The proof is not given as it follows step by step, with minor modifications, that of [18, Sections 2–3].

Finally, note that Q-superlinear convergence would follows if Assumption 6 were replaced with the stronger assumption

$$\lim_{k\to\infty} \frac{\|P_k(H_k - \nabla^2_{xx} L(x^*, \lambda^*)) d_k\|}{\|d_k\|} = 0.$$

(See, e.g., [2].)

4 Implementation and Numerical Results

Our implementation of **FSQP′** (in C) differs in a number of ways from the algorithm stated in Section 2. (It is readily checked that none of the differences significantly affect the convergence analysis of Section 3.) Just like in existing C implementation of **FSQP** (CFSQP: see [12]) the distinctive character of linear (affine) constraints and of simple bounds is exploited (provided the nature of these constraints is made explicit). Thus the general form of the problem description tackled by our implementation is

$$\begin{array}{lll} \min & f(x) & \\ \text{s.t.} & g_j(x) \le 0, & j = 1, \ldots, m_n, \\ & \langle a_j, x\rangle + b_j \le 0, & j = 1, \ldots, m_a, \\ & x^\ell \le x \le x^u, & \end{array}$$

where $a_j \in \mathbb{R}^n$, $b_j \in \mathbb{R}$, $j = 1, \ldots, m_a$, and x^ℓ, $x^u \in \mathbb{R}^n$ with $x^\ell < x^u$ (componentwise). The details of the implementation are spelled out below. Many of them, including the update rule for H_k, are exactly as in CFSQP.

In the implementation of $QP(x_k, H_k, \eta_k)$, no "tilting" is effected in connection with the linear constraints and simple bound, since clearly the untilted SQP direction is feasible for these constraints. In addition, each nonlinear constraint is assigned its own tilting parameter η_k^j, $j = 1, \ldots, m_n$. Thus $QP(x_k, H_k, \eta_k)$ is replaced with

$$\begin{array}{lll} \min & \frac{1}{2}\langle d, H_k d\rangle + \gamma & \\ \text{s.t.} & \langle \nabla f(x_k), d\rangle \le \gamma, & \\ & g_j(x) + \langle \nabla g_j(x), d\rangle \le \gamma \cdot \eta_k^j, & j = 1, \ldots, m_n, \\ & \langle a_j, x_k + d\rangle + b_j \le 0, & j = 1, \ldots, m_a, \\ & x^\ell - x_k \le d \le x^u - x_k. & \end{array}$$

The η_k^j's are updated independently, based on independently adjusted C_k^j's. In the algorithm description and in the analysis all that was required of C_k was that it remain bounded and bounded away from zero. In practice, though, performance of the algorithm is critically dependent upon the choice of C_k. In the implementation, an adaptive scheme was chosen in which the new values C_{k+1}^j are selected in Step 3 based on their previous values C_k^j, on the outcome of the arc search in Step 2, and on a preselected parameter $\delta_c > 1$. Specifically: (i) if the full step of one was accepted ($t_k = 1$), then all C^j are left unchanged; (ii) if the step of one was not accepted even though all trial points were feasible, then, for all j, C_k^j is decreased to $\min\{\delta_c C_k^j, \overline{C}\}$; (iii) if some infeasibility was encountered in the arc search, then, for all j such that g_j caused a step reduction at some trial point, C_k^j is increased to $\max\{C_k^j/\delta_c, \underline{C}\}$ and, for all other j, C_k^j is kept constant. Here, g_j is said to cause a step reduction if, for some trial point x, g_j is violated (i.e., $g_j(x) > 0$) but all constraints checked at x before g_j were found to be satisfied at that point. (See below for the order in which constraints are checked in the arc search.)

It was stressed in Section 2 that the Maratos correction can be computed using an inequality-constrained QP such as QP^{C}, instead of a LS^{C}. This was done in our numerical experiments, in order to more meaningfully compare the new algorithm with CFSQP, in which an in an inequality-constrained QP is indeed used. The implementation of QP^{C} and LS^{E} involves index sets of "almost active" constraints and of binding constraints. First we define

$$I_k^n = \{\ j \mid g_j(x_k) + \langle \nabla g_j(x_k), d_k \rangle - \gamma_k \cdot \eta_k^j > -\sqrt{\epsilon_m}\ \}$$
$$I_k^a = \{\ j \mid \langle a_j, x_k + d_k \rangle + b_j > -\sqrt{\epsilon_m}\ \}$$

where ϵ_m is the machine precision. Next, the *binding* sets are defined as

$$I_k^{b,n} = \{\ j \mid \lambda_k^j > 0\ \}, \qquad I_k^{b,a} = \{\ j \mid \lambda_k^{a,j} > 0\ \},$$
$$I_k^{b,l} = \{\ j \mid \zeta_k^{l,j} > 0\ \}, \qquad I_k^{b,u} = \{\ j \mid \zeta_k^{u,j} > 0\ \}.$$

where $\lambda_k \in \mathbb{R}^{m_n}$ is now the QP multiplier corresponding to the nonlinear constraints and where $\lambda_k^a \in \mathbb{R}^{m_a}$, $\zeta_k^u \in \mathbb{R}^n$, and $\zeta_k^l \in \mathbb{R}^n$ are the QP multipliers corresponding to the affine constraints, the upper bounds, and the lower bounds, respectively. Of course, no bending is required from d_k^{C} in connection with affine constraints and simple bounds, hence if $I_k^n = \emptyset$, we simply set $d_k^{\mathrm{C}} = 0$. Otherwise the following modification of QP^{C} is used:

$$\begin{array}{lll}
\min & \langle d_k + d^{\mathrm{C}}, H_k(d_k + d^{\mathrm{C}}) \rangle + \langle \nabla f(x_k), d_k + d^{\mathrm{C}} \rangle & \\
\text{s.t.} & g_j(x_k + d_k) + \langle \nabla g_j(x_k), d^{\mathrm{C}} \rangle \le -\min\{10^{-2}\|d_k\|, \|d_k\|^\tau\}, & j \in I_k^n, \\
& \langle a_j, x_k + d_k + d^{\mathrm{C}} \rangle + b_j \le 0, & j \in I_k^a, \\
& d^{\mathrm{C},j} \le x^u - x_k^j - d_k^j, & j \in I_k^{b,u}, \\
& d^{\mathrm{C},j} \ge x^l - x_k^j - d_k^j, & j \in I_k^{b,l}.
\end{array}$$

Since not all simple bounds are included in the computation of d_k^{C}, it is possible that $x_k + d_k + d_k^{\mathrm{C}}$ will not satisfy all bounds. To take care of this, we simply "clip" d_k^{C} so that the bounds are satisfied. Specifically, for the upper bounds, we perform the following:

$$
\begin{array}{l}
\textbf{for} \;\; j \notin I_k^{b,u} \;\; \textbf{do} \\
\quad \textbf{if} \;\; (d_k^{\mathrm{C},j} \geq x^u - x_k^j - d_k^j) \;\; \textbf{then} \\
\qquad d_k^{\mathrm{C},j} \leftarrow x^u - x_k^j - d_k^j \\
\textbf{end}
\end{array}
$$

The same procedure, mutatis mutandis, is executed for the lower bounds. We note that such a procedure has no effect on the convergence analysis of Section 3 since, locally, the active set is correctly identified and a full step along $d_k + d_k^{\mathrm{C}}$ is always accepted. The least squares problem LS^{E} used to compute d_k^{E} is modified similarly. Specifically, in the implementation, d_k^{E} is only computed if $m_n > 0$, in which case we use

$$
\begin{array}{lll}
\min & \frac{1}{2}\langle d^{\mathrm{E}}, H_k d^{\mathrm{E}} \rangle + \langle \nabla f(x_k), d^{\mathrm{E}} \rangle & \\
\text{s.t.} & g_j(x_k) + \langle \nabla g_j(x_k), d^{\mathrm{E}} \rangle = 0, & j \in I_{k-1}^{b,n}, \\
& \langle a_j, x_k + d^{\mathrm{E}} \rangle + b_j = 0, & j \in I_{k-1}^{b,a}, \\
& d^{\mathrm{E},j} = x^u - x_k^j, & j \in I_{k-1}^{b,u}, \\
& d^{\mathrm{E},j} = x^l - x_k^j, & j \in I_{k-1}^{b,l}.
\end{array}
$$

The implementation of the arc search (Step 2) is as in CFSQP. Specifically, feasibility is checked before sufficient decrease, and testing at a trial point is aborted as soon as infeasibility is detected. Like in CFSQP, all linear and bound constraints are checked first, then nonlinear constraints in an order maintained as follows: (i) at the start of the arc search from a given iterate x_k the order is reset to be the natural numerical order; (ii) within an arc search, as a constraint is found to be violated at a trial point, its index is moved to the beginning of the list, with the order of the others left unchanged.

An aspect of the algorithm which was intentionally left vague in Sections 2 and 3 was the updating scheme for the Hessian estimates H_k. In the implementation, we use the BFGS update with Powell's modification [19]. Specifically, define

$$
\delta_{k+1} \triangleq x_{k+1} - x_k
$$

$$
y_{k+1} \triangleq \nabla_x L(x_{k+1}, \lambda_k) - \nabla_x L(x_k, \lambda_k),
$$

where, in an attempt to better approximate the true multipliers, if $\mu_k > \sqrt{\epsilon_m}$ we normalize as follows

$$\lambda_k^j \leftarrow \frac{\lambda_k^j}{\mu_k}, \quad j = 1, \ldots, m_n.$$

A scalar $\theta_{k+1} \in (0, 1]$ is then defined by

$$\theta_{k+1} \triangleq \begin{cases} 1, & \text{if } \langle \delta_{k+1}, y_{k+1} \rangle \geq 0.2 \cdot \langle \delta_{k+1}, H_k \delta_{k+1} \rangle, \\ \dfrac{0.8 \cdot \langle \delta_{k+1}, H_k \delta_{k+1} \rangle}{\langle \delta_{k+1}, H_k \delta_{k+1} \rangle - \langle \delta_{k+1}, y_{k+1} \rangle}, & \text{otherwise.} \end{cases}$$

Defining $\xi_{k+1} \in \mathbb{R}^n$ as

$$\xi_{k+1} \triangleq \theta_{k+1} \cdot y_{k+1} + (1 - \theta_{k+1}) \cdot H_k \delta_{k+1},$$

the rank two Hessian update is

$$H_{k+1} = H_k - \frac{H_k \delta_{k+1} \delta_{k+1}^T H_k}{\langle \delta_{k+1}, H_k \delta_{k+1} \rangle} + \frac{\xi_{k+1} \xi_{k+1}^T}{\langle \delta_{k+1}, \xi_{k+1} \rangle}.$$

Note that while it is not clear whether the resultant sequence $\{H_k\}$ will, in fact, satisfy Assumption 6, this update scheme is known to perform very well in practice.

All QPs and linear least squares subproblems were solved using QPOPT [7]. For comparison sake, QPOPT was also used to solve the QP subproblems in CFSQP. While the default QP solver for CFSQP is the public domain code QLD (see [22]), we opted for QPOPT because it allows "warm starts" and thus is fairer to CFSQP in the comparison with the implementation of **FSQP′** (since more QPs are solved with the former). For alls QPs in both codes, the active set in the solution at a given iteration was used as initial guess for the active set for the same QP at the next iteration.

In order to guarantee that the algorithm terminates after a finite number of iterations with an approximate solution, the stopping criterion of *Step 1* is changed to

$$\textbf{if} \ \ (\|d_k\| \leq \epsilon) \ \ \textbf{stop}, \tag{23}$$

where $\epsilon > 0$ is small. Finally, the following parameter values were selected:

$$\begin{array}{lll} \alpha = 0.1, & \beta = 0.5, & \tau = 2.5, \\ \epsilon_\ell = \sqrt{\epsilon}, & \underline{C} = 1 \times 10^{-3}, & \overline{C} = 1 \times 10^3, \\ \delta_c = 2, & \bar{D} = 10 \cdot \epsilon_\ell. & \end{array}$$

Further, we always set $H_0 = I$, and $C_0^j = 1$ and $\eta_0^j = \epsilon C_0^j (= \epsilon)$, $j = 1, \ldots, m_n$. All experiments were run on a Sun Microsystems Ultra 5 workstation.

For the first set of numerical tests, we selected a number of problems from [9] which provided feasible initial points and contained no equality constraints. The results are reported in Table 1, where the performance of our implementation of **FSQP′** is compared with that of CFSQP (with QPOPT as QP solver). The column labeled # lists the problem number as given in [9], the column labeled `ALGO` is self-explanatory. The next three columns give the size of the problem following the conventions of this section. The columns labeled `NF`, `NG`, and `IT` give the number of objective function evaluations, nonlinear constraint function evaluations, and iterations required to solve the problem, respectively. Finally, $f(x^*)$ is the objective function value at the final iterate and ϵ is as above. The value of ϵ was chosen in order to obtain approximately the same precision as reported in [9] for each problem.

The results reported in Table 1 are encouraging. The performance of our implementation of Algorithm **FSQP′** in terms of number of iterations and function evaluations is essentially identical to that of CFSQP (Algorithm **FSQP**). The expected payoff of using **FSQP′** instead of **FSQP** however is that, on large problems the CPU time expended in linear algebra, specifically in solving the QP and linear least squares subproblems, should be much less. To assess this, we carried out comparative tests on the COPS suite of problems [3].

The first five problems from the COPS set [3] were considered, as these problems either do not involve nonlinear equality constraints or are readily reformulated without such constraints. (Specifically, in problem "Sphere" the equality constraint was changed to a "$\leq$" constraint; and in "Chain" the equality constraint (with $L = 4$) was replaced with two inequalities, with the left-hand side constrained to be between the values $L = 4$ and $L = 5$; the solution was always at 5.) "Sawpath" was discarded because it involves few variables and many constraints, which is not the situation at which the new algorithm is targeted. The results obtained with various instances of the other four problems are presented in Table 2. The format of that table is identical to that of Table 1 except for the additional column labeled NQP. In that column we list the total number of QP iterations in the solution of the two major QPs, as reported by QPOPT. (Note that QPOPT reports

#	ALGO	n	m_a	m_n	NF	NG	IT	$f(x^*)$	ϵ
12	**FSQP′**	2	0	1	7	14	7	-3.0000000E+01	1.E–6
	CFSQP				7	14	7	-3.0000000E+01	
29	**FSQP′**	3	0	1	11	20	10	-2.2627417E+01	1.E–5
	CFSQP				11	20	10	-2.2627417E+01	
30	**FSQP′**	3	0	1	18	35	18	1.0000000E+00	1.E–7
	CFSQP				18	35	18	1.0000000E+00	
31	**FSQP′**	3	0	1	9	36	8	6.0000000E+00	1.E–5
	CFSQP				9	19	7	6.0000000E+00	
33	**FSQP′**	3	0	2	4	11	4	-4.0000000E+00	1.E–8
	CFSQP				4	11	4	-4.0000000E+00	
34	**FSQP′**	3	0	2	8	34	8	-8.3403245E–01	1.E–8
	CFSQP				7	28	7	-8.3403244E–01	
43	**FSQP′**	4	0	3	9	51	9	-4.4000000E+01	1.E–5
	CFSQP				10	46	8	-4.4000000E+01	
66	**FSQP′**	3	0	2	8	30	8	5.1816327E–01	1.E–8
	CFSQP				8	30	8	5.1816327E–01	
84	**FSQP′**	5	0	6	4	37	4	-5.2803351E+06	1.E–8
	CFSQP				4	30	4	-5.2803351E+06	
93	**FSQP′**	6	0	2	13	54	12	1.3507596E+02	1.E–5
	CFSQP				16	62	13	1.3507596E+02	
113	**FSQP′**	10	3	5	12	120	12	2.4306210E+01	1.E–3
	CFSQP				12	108	12	2.4306377E+01	
117	**FSQP′**	15	0	5	20	205	19	3.2348679E+01	1.E–4
	CFSQP				20	219	19	3.2348679E+01	

Table 1: Numerical results on Hock-Schittkowski problems.

zero iteration when the result of the first step onto the working set of linear constraints happens to be optimal. To be "fair" to **FSQP′** , we thus do not count the work involved in solving LS^{E} either. We also do not count the QP iterations in solving QP^{C}, the "correction" QP, because it is invoked identically in both algorithms.) The reason for the smaller ϵ on `Cam` is this allowed CFSQP to reach the globally optimal objective function values (as per [3]).

The results show a typical significantly lower number of QP iterations with the new algorithm and, as in the case of the Hock-Schittkowski problems, a roughly comparable behavior of the two algorithms in terms of number of function evaluations. Note that in the two instances where the NQP count is less for CFSQP than for $\mathbf{FSQP'}$, different local minima are reached, which makes the comparison meaningless. Finally, the abnormal terminations on `Sphere-50` and `Sphere-100` are both due to QPOPT's failure to solve a QP—the "tilting" QP in the case of CFSQP.

5 Discussion

One issue of interest is whether our convergence results still hold under weaker assumptions. To wit, Qi and Wei showed in [20] that the algorithm of [15] still enjoys global convergence (all limit points are KKT) and local (two-step) superlinear convergence when Assumption 3 (LICQ) is replaced with the Mangasarian-Fromovitz constraint qualification (MFCQ)—or even with a condition slightly weaker than MFCQ. They further showed that, if that algorithm is slightly modified, local superlinear convergence is preserved without strict complementarity assumption, provided the strong second-order sufficiency condition (SSOSC) is assumed. For algorithm $\mathbf{FSQP'}$ as stated however, LICQ and strict complementarity are essential, in connection with Step 3 (iii). First, Assumption 3 is needed in order for $LS^{\mathrm{E}}(x_k, H_k, I_k^{\mathrm{E}})$ to be well-defined close to a solution x^*. Second, strict positivity of the multipliers associated with active constraints at the solution is needed in order for the components of λ_k^{E} to be nonnegative when a solution is approached. Barring this, the condition $\lambda_{k+1}^{\mathrm{E}} \geq 0$ in Step 3(iii) may never hold and the update rule $\eta_{k+1} \leftarrow C_{k+1} \cdot \|d_k\|^2$ may not be used close to the solution, in which case superlinear convergence would not take place. Careful modifications of Algorithm $\mathbf{FSQP'}$ might (at least in theory) accommodate weaker assumptions though. First, in the absence of Assumption 3, $LS^{\mathrm{E}}(x_k, H_k, I_k^{\mathrm{E}})$ could possibly be replaced with a QP with $|I_k^{\mathrm{E}}|$ inequality constraints (with essentially no penalty in CPU cost). Second, by replacing the nonnegativity condition by a requirement of the type $\lambda^{\mathrm{E},j} \geq -\epsilon_k$, where $\lambda^{\mathrm{E},j}$ is now a "regular" multiplier (see [20]) and where $\epsilon_k > 0$ would be made to go to zero at an appropriate (slow) rate, it may be possible to preserve convergence to KKT points while insuring that, even in the absence of strict complementarity, the test would be satisfied close to the solution, thus allowing superlinear convergence to take place. This would require detailed analysis though, and may be as likely to hurt as to help in practice.

A second issue worth discussing is that of possible low cost solution for the QP and two linear least-squares problems. The indexes of constraints appearing in LS^{E} and LS^{C} at iteration k are generally different, I_k^{E} $(=I_{k+1})$

P	ALGO	n	m_a	m_n	NF	NG	IT	NQP	$f(x^*)$	ϵ
Polygon-10	**FSQP′**	18	8	36	17	798	18	51	.749137	1.E–4
	CFSQP				16	740	18	91	.749137	
Polygon-20	**FSQP′**	38	18	171	27	5552	28	142	.776859	1.E–4
	CFSQP				42	8177	44	350	.776859	
Polygon-40	**FSQP′**	78	38	741	267	208706	107	571	.783062	1.E–4
	CFSQP				243	126592	106	1689	.783062	
Polygon-50	**FSQP′**	98	48	1176	1023	1232889	273	938	.783062	1.E–4
	CFSQP				591	345458	154	2771	.783873	
Sphere-20	**FSQP′**	60	0	20	1462	35114	280	302	150.882	1.E–4
	CFSQP				1812	20920	352	745	150.882	
Sphere-30	**FSQP′**	90	0	30	8318	280532	1016	1065	359.604	1.E–4
	CFSQP				6494	74797	837	1743	359.604	
Sphere-40	**FSQP′**	120	0	40	1445	70960	311	406	660.675	1.E–4
	CFSQP				795	28328	246	587	660.675	
Sphere-50	**FSQP′**	150	0	50					failure	1.E–4
	CFSQP				2300	80467	560	1568	1055.18	
Sphere-100	**FSQP′**	300	0	50	516	119252	506	3589	4456.06	1.E–4
	CFSQP								failure	
Chain-50	**FSQP′**	50	0	2	247	1165	189	203	4.81198	1.E–4
	CFSQP				274	1080	199	403	4.81198	
Chain-100	**FSQP′**	100	0	2	616	2793	359	396	4.81190	1.E–4
	CFSQP				633	2187	369	768	4.81190	
Chain-150	**FSQP′**	150	0	2	1454	5522	586	654	4.81189	1.E–4
	CFSQP				977	3784	575	1259	4.81189	
Chain-200	**FSQP′**	200	0	2	1544	6772	710	819	4.81189	1.E–4
	CFSQP				1439	5326	759	1716	4.81188	
Cam-50	**FSQP′**	50	1	102	13	7347	39	223	-214.761	1.E–5
	CFSQP				12	6288	39	604	-214.761	
Cam-100	**FSQP′**	100	1	202	32	28896	78	591	-428.096	1.E–5
	CFSQP				14	21558	61	1322	-428.415	
Cam-200	**FSQP′**	200	1	402	65	107905	146	2961	-844.319	1.E–5
	CFSQP				18	74352	100	2929	-855.700	
Cam-400	**FSQP′**	400	1	802	54	296080	194	10331	-1654.45	1.E–5
	CFSQP				21	244795	161	6232	-1710.28	

Table 2: Numerical results on COPS problems.

for the former, I_k for the latter. However, it was proved that, for k large enough, both of these sets are equal to $I(x^*)$. When that is the case, it is readily checked that $LS^{\mathrm{E}}(x_k, H_k, I_k^{\mathrm{E}})$ and $LS^{\mathrm{C}}(x_k, d_k, H_k, I_k)$ involve the same matrix, and thus that the latter can be solved at low cost once the former has been solved. Unfortunately, the linear systems arising in the solution of $QP(x_k, H_k, \eta_k)$ are different. In particular, they involve η_k.

6 Conclusions

We have presented here a new SQP-type algorithm generating feasible iterates. The main advantage of the algorithm presented here is a reduction in the amount of computation required in order to generate a new iterate. While this may not be very important for applications where function evaluations dominate the actual amount of work to compute a new iterate, it is very useful in many contexts. In any case, we saw in the previous section that preliminary results seem to indicate that decreasing the amount of computation per iteration did not come at the cost of increasing the number of function evaluations required to find a solution.

A number of significant extensions of Algorithm **FSQP′** are being examined. It is not too difficult to extend the algorithm to handle mini-max problems. The only real issue that arises is how to handle the mini-max objectives in the least squares sub-problems. Several possibilities, each with the desired global and local convergence properties, are being examined. Another extension that is important for engineering design is the incorporation of a scheme to efficiently handle very large sets of constraints and/or objectives. We will examine schemes along the lines of those developed in [11, 25]. Further, work remains to be done to exploit the close relationship between the two least squares problems and the quadratic program. A careful implementation should be able to use these relationships to great advantage computationally. For starters, updating the Cholesky factors of H_k instead of H_k itself at each iteration would save a factorization in each of the sub-problems. Finally, it is possible to extend the class of problems (P) which are handled by the algorithm to include nonlinear equality constraints. Of course, we will not be able to generate feasible iterates for such constraints, but a scheme such as that studied in [10] could be used in order to guarantee asymptotic feasibility while maintaining feasibility for all inequality constraints.

References

[1] J. Birge, L. Qi, and Z. Wei. A variant of the Topkis-Veinott method for solving inequality constrained optimization problems. Technical Report AMR 97/29, School of Mathematics, The University of New South Wales, Sydney, Australia, 1997.

[2] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numerica*, pages 1–51, 1995.

[3] A.S. Bondarenko, D.M. Bortz, , and J.J. Moré. COPS, Large-scale nonlinearly constrained optimization problems. Technical Report ANL/MCS-TM-237, Argonne National Laboratory, Argonne, Illinois 60439, 1999.

[4] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large scale nonlinear programming. Technical Report 97-05, Optimization Technology Center, Argonne National Laboratory, 1997.

[5] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang. On the formulation and theory of the Newton interior-point method for nonlinear programming. *J. Opt. Theory Appl.*, 89:507–541, 1996.

[6] D. M. Gay, M. L. Overton, and M. H. Wright. A primal-dual interior method for nonconvex nonlinear programming. Technical Report 97-4-08, Bell Laboratories, Computing Sciences Research Center, 1997.

[7] P.E. Gill, W. Murray, and M.A. Sounders. User's guide for qpopt 1.0: A fortran package for quadratic programming. Technical report, Stanford University, 1995.

[8] J. N. Herskovits and L. A. V. Carvalho. A successive quadratic programming based feasible directions algorithm. In A. Bensoussan and J.L. Lions, editors, *Proc. of the Seventh International Conference on Analysis and Optimization of Systems — Antibes, June 25-27, 1986, Lecture Notes in Control and Information Sciences 83*, pages 93–101. Springer Verlag, Berlin, 1986.

[9] W. Hock and K. Schittkowski. *Test Examples For Nonlinear Programming Codes, Lecture Notes in Economics and Mathematical Systems No. 187.* Springer-Verlag, Berlin, 1981.

[10] C. T. Lawrence and A. L. Tits. Nonlinear equality constraints in feasible sequential quadratic programming. *Optimization Methods and Software*, 6:265–282, 1996.

[11] C. T. Lawrence and A. L. Tits. Feasible sequential quadratic programming for finely discretized problems from SIP. In R. Reemtsen and J.-J. Rückmann, editors, *Semi-infinite Programming*, pages 159–193. Kluwer Academic Publishers B.V., 1998. In the series Nonconvex Optimization and its Applications.

[12] C. T. Lawrence, J. L. Zhou, and A. L. Tits. *User's Guide for CFSQP Version 2.5: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints*, 1997. Institute for Systems Research, University of Maryland (College Park, MD).

[13] N. Maratos. *Exact Penalty Functions for Finite Dimensional and Control Optimization Problems.* PhD thesis, Imperial College of Science and Technology, 1978.

[14] E. R. Panier and A. L. Tits. A superlinearly convergent feasible method for the solution of inequality constrained optimization problems. *SIAM Journal on Control and Optimization*, 25(4):934–950, 1987.

[15] E. R. Panier and A. L. Tits. On combining feasibility, descent and superlinear convergence in inequality constrained optimization. *Math. Programming*, 59:261–276, 1993.

[16] E. R. Panier, A. L. Tits, and J. N. Herskovits. A QP-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization. *SIAM J. Control and Optimization*, 26(4):788–811, 1988.

[17] E. Polak. *Computational Methods in Optimization.* Academic Press, New York, 1971.

[18] M. J. D. Powell. Convergence of variable metric methods for nonlinearly constrained optimization calculations. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, editors, *Nonlinear Programming 3*, pages 27–63. Academic Press, New York, 1978.

[19] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In G. A. Watson, editor, *Numerical Analysis, Dundee,*

1977, Lecture Notes in Mathematics 630, pages 144–157. Springer Verlag, 1978.

[20] L. Qi and Z. Wei. On the constant positive linear dependence condition and its application to SQP methods. Technical Report AMR 97/4, School of Mathematics, The University of New South Wales, Sydney, Australia, 1997. To appear in SIAM J. Optim.

[21] S. M. Robinson. Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear-programming algorithms. *Math. Programming*, 7:1–16, 1974.

[22] K. Schittkowski. *QLD: A Fortran Code for Quadratic Programming, User's Guide.* Mathematisches Institut, Universität Bayreuth, Germany, 1986.

[23] T. Urban, A. L. Tits, and C. T. Lawrence. A primal-dual interior-point method for nonconvex optimization with multiple logarithmic barrier parameters and with strong convergence properties. Technical Report 98-27, Institute for Systems Research, University of Maryland, College Park, 1998.

[24] R. J. Vanderbei and D. F. Shanno. An interior point algorithm for nonconvex nonlinear programming. Technical Report SOR-97-21, Princeton University, Dept. of Statistics and Operations Research, 1997.

[25] J. L. Zhou and A. L. Tits. An SQP algorithm for finely discretized continuous minimax problems and other minimax problems with many objective functions. *SIAM J. on Optimization*, pages 461–487, May 1996.

[26] J. L. Zhou, A. L. Tits, and C. T. Lawrence. *User's Guide for FSQP Version 3.7: A FORTRAN Code for Solving Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality and Linear Constraints*, 1997. ISR TR-92-107r2, Institute for Systems Research, University of Maryland (College Park, MD).

[27] G. Zoutendijk. *Methods of Feasible Directions.* Elsevier Science, Amsterdam, The Netherlands, 1960.